EXHIBIT 1
Part 1 of 4

UNITED STATES PATENT AND TRADEMARK OFFICE

_____

BEFORE THE PATENT TRIAL AND APPEAL BOARD

_____

ARISTA NETWORKS, INC.
Petitioner

v.

CISCO SYSTEMS, INC.
Patent Owner

_____

Case No. <u>IPR2016-00119</u>
Patent No. 7,047,526

_____

**PETITION FOR *INTER PARTES* REVIEW
OF U.S. PATENT NO. 7,047,526**

Mail Stop PATENT BOARD Patent Trial and Appeal Board U.S. Patent & Trademark Office P.O. Box 1450 Alexandria, VA 22313-1450	Submitted by: Keker & Van Nest LLP 633 Battery Street San Francisco, CA 94111 (415) 391-5400 (tel) (415) 397-7188 (fax) Counsel for Petitioner Arista Networks, Inc. November 4, 2015

Petition for Inter Partes Review of Patent No. 7,047,526

TABLE OF CONTENTS

Petition for Inter Partes Review of Patent No. 7,047,526

iii

Petition for Inter Partes Review of Patent No. 7,047,526

Petition for Inter Partes Review of Patent No. 7,047,526

Petition for Inter Partes Review of Patent No. 7,047,526

Petition for Inter Partes Review of Patent No. 7,047,526

Petition for Inter Partes Review of Patent No. 7,047,526

TABLE OF AUTHORITIES

Petition for Inter Partes Review of Patent No. 7,047,526

Petition for Inter Partes Review of Patent No. 7,047,526

EXHIBIT LIST

Exh. No.	Description
1001	U.S. Patent No. 7,047,526 to Wheeler et al., issued May 16, 2006 ("'526 Patent").
1002	U.S. Patent No. 6,523,172 to Martinez-Guerra et al., issued Feb. 18, 2003 ("Martinez-Guerra").
1003	Noam Chomsky & George A. Miller, *Finite State Language*, 1 Information and Control 91-112 (1958).
1004	Noam Chomsky, *On Certain Formal Properties of Grammars*, 2 Information and Control 137-167 (1959).
1005	Alfred V. Aho, et. al., Compilers: Principles, Techniques, And Tools (1986).
1006	Gerry Kane, MIPS RISC Architecture (1987).
1007	OpenVMS User's Manual, version 7.1 (Nov. 1996), http://www.mi.infn.it/~calcolo/OpenVMS/ssb71/6489/6489p.htm.
1008	Honeywell Bull, Multics: Commands and Active Functions (1985).
1009	Brian W. Kernighan & Rob Pike, The UNIX Programming Environment (1984).
1010	Cisco's Preliminary Claim Construction Disclosure, *Cisco Systems, Inc. v. Arista Networks, Inc.*, No. 5:14-cv-05344-BLF (N.D. Cal. Aug. 24, 2015).
1011	Robert Sedgwick, Algorithms in C (3rd ed. 1998).
1012	Dennis M. Ritchie & Ken Thompson, The UNIX Time-Sharing System, 17 Communications of the ACM 365 (Jul. 1974).
1013	U.S. Patent No. 6,134,709 to Pratt, issued Oct. 17, 2000.
1014	Declaration of Douglas W. Clark, Ph.D.

Petition for Inter Partes Review of Patent No. 7,047,526

I.   INTRODUCTION

Arista Networks Inc. ("Petitioner") petitions for *inter partes* review of

claims 1-26 ("the challenged claims") of United States Patent No. 7,047,526 to

Wheeler *et al.,* entitled "Generic Command Interface for Multiple Executable

Routines" ("'526 patent") (Ex. 1001).

The '526 patent is directed to a system and method for parsing "generic

commands" received from a user and translating them into "prescribed commands"

that are issued to a selected "management program." (*Id.*, Abstract.) The

specification explains that "system administrators may attempt to utilize multiple

tools within a software system . . . for improved system performance." (*Id.* at 1:28-

31.) A problem with using many tools, however, is that it "requires the users to

remember the names and syntaxes of numerous commands for the respective . . .

programs and . . . tools." (*Id.* at 1:32-34.)

The '526 patent purports to solve this problem by providing a system and

method for integrating multiple tools and programs "without the necessity of

learning the respective command formats and syntax." (*Id.* at 1:43-44.) It claims a

system that includes a "command parse tree" configured to parse a "generic

command" received from a user. (*Id.* at 1:50-54.) According to the specification, a

"generic command" is "an abstraction of the tool-specific command formats and

Petition for Inter Partes Review of Patent No. 7,047,526

syntax, enabling a user to issue command[s] based on the relative functions, as opposed to the specific syntax for a corresponding tool." (*Id.* at 3:32-35.)

"The command parse tree includes multiple elements, each specifying at least one corresponding generic command component and a corresponding at least one command action value." (*Id.* at 1:51-54.) The parser identifies the element that's the "best match" for the generic command, then "issues a prescribed command for a selected one of the management programs according to [its] corresponding command format based on the selected command action value." (*Id.* at 1:54-58.) According to the patent, this allows a user to "control multiple management programs having respective command formats, by using a set of generic commands that are independent from the command formats, eliminating the necessity that the user needs to learn the detailed command formats and syntax." (*Id.* at 1:58-63.)

The '526 patent claimed nothing new when it was applied for in June 2000. The idea of translating commands from a high-level format into a tool-specific format for a particular component is fundamental to computer science, and had been known for decades. To cite just one of many examples, Digital Equipment Corporation's "Digital Command Language," or "DCL," was used since the 1980s to translate high-level commands entered by a user (*e.g.*, "PRINT/COPIES = 5 GROCERY.LIS," to print five copies of a grocery list file) into specialized

2

Petition for Inter Partes Review of Patent No. 7,047,526

commands issued to external programs. (*See* Declaration of Douglas Clark, Ph.D ("Clark Decl.") (Ex. 1014) at ¶¶ 25; *see also* Ex. 1007 at 18-19, Section 3.3.1.) Using a "command parse tree" to parse and translate the high-level commands, as recited in the '526 patent, was equally obvious.

This petition relies on prior-art United States Patent No. 6,523,172 ("Martinez-Guerra") (Ex. 1002). Martinez-Guerra is directed to a "parser-translator technology" that translates statements from "a high-level user language" into ones that are "appropriate to a particular data processing application," so that "a user can focus on the semantics of the desired operations and need not be concerned with the proper syntax of a language for a particular system." (*Id.*, Abstract.) As explained below, and in the accompanying Declaration of Dr. Douglas Clark, Martinez-Guerra discloses nearly every limitation of every challenged claim of the '526 patent. And where Martinez-Guerra arguably does not disclose a claim limitation, any necessary modification was well known at the time of the alleged invention and obvious to a person of ordinary skill in the art.

For these reasons, the Board should institute IPR of all the challenged claims, and should find them invalid on the grounds set forth below.

II.   MANDATORY NOTICES (37 C.F.R. § 42.8(a)(1))

Pursuant to 37 C.F.R. § 42.8, Petitioner provides the following mandatory disclosures:

3

Petition for Inter Partes Review of Patent No. 7,047,526

A.   Real Parties-In-Interest (37 C.F.R. § 42.8(b)(1))

The real party-in-interest is Arista Networks, Inc.

B.   Related Matters (37 C.F.R. § 42.8(b)(2))

The '526 patent is at issue in *Cisco Systems, Inc. v. Arista Networks, Inc.*,

No. 5:14-cv-05344 (N.D. Cal).

C.   Designation of Lead and Back-Up Counsel (37 C.F.R. § 42.8(b)(3))

Petitioner appoints Eugene M. Paige (Reg. No. 55,519) of Keker & Van

Nest LLP as lead counsel, and appoints Robert A. Van Nest (*pro hac vice* motion

to be filed), Brian L. Ferrall (*pro hac vice* motion to be filed), and David J. Silbert

(*pro hac vice* motion to be filed) of Keker & Van Nest LLP as back-up counsel. An

appropriate Power of Attorney is filed concurrently herewith.

D.   Service Information (37 C.F.R. § 42.8(b)(4))

Service of any documents to lead or back-up counsel may be made to:

> Keker & Van Nest LLP
> 633 Battery Street
> San Francisco, CA 94111
> (415) 391-5400 (telephone); (415) 397-7188 (fax)

Petitioner consents to service by email at the following addresses:

aristaipr@kvn.com; epaige@kvn.com; dsilbert@kvn.com.

III.   STANDING (37 C.F.R. § 42.104(a))

Petitioner certifies that (1) the '526 patent is available for *inter partes*

review; and (2) Petitioner is not barred or estopped from requesting *inter partes*

Petition for Inter Partes Review of Patent No. 7,047,526

review of any claim of the '526 patent on the grounds identified herein. This petition is filed in accordance with 37 C.F.R. § 42.106(a). Concurrently filed herewith is an Exhibit List per 37 C.F.R. § 42.63(e). The Office is authorized to charge the required fee as set forth in 37 C.F.R. § 42.15(a) to Deposit Acct. No. 506260. The Office is further authorized to charge fee deficiencies and credit overpayments to the above-referenced Deposit Account.

IV.   OVERVIEW

A.   The alleged invention of the '526 patent

The '526 patent describes an alleged improvement for controlling complex administration and/or diagnostic software tools in processor-based systems. (Ex. 1001 at 1:11-14.) It explains that, typically, each of these administration and diagnostic tools has its own command format, which system administrators must remember. (*Id.* at 1:31-34.) Moreover, system administrators may find it difficult "to determine which tool is the best tool (and/or which is the best syntax) to use for a given problem." (*Id.* at 3:16-20.)

To address these problems, the patent proposes a set of "generic commands" that a user can input into a parser-translator system, which will issue a corresponding "prescribed command" to a management program in that program's command format. (*See id.*, Abstract.) As the patent explains, "the new syntax provides a generic instruction set that provides an abstraction of the tool-specific

Petition for Inter Partes Review of Patent No. 7,047,526

command formats and syntax, enabling the user to issue command[s] based on the relative functions, as opposed to the specific syntax for a corresponding tool . . . ."

(*Id.* at 3:31-35; *see also id.* at 7:1-9:16 (listing "Generic Command Examples").)

The generic command is validated by the system in relation to what the patent calls a "command parse tree." (*Id.* at 1:48-51.) Figure 2, which is reproduced below, depicts the command parse tree (structure 22) of the preferred embodiment:
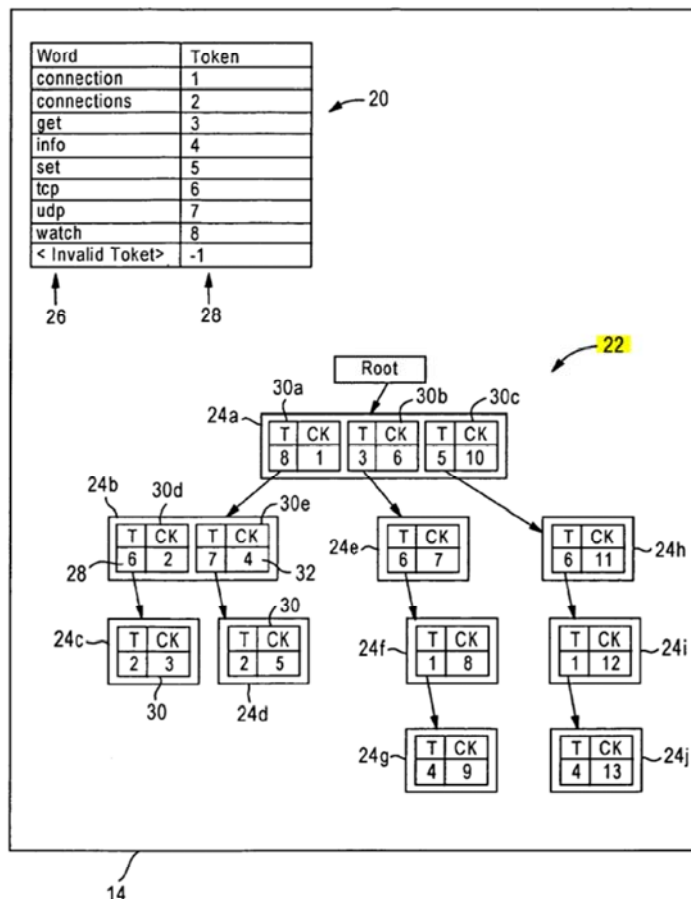
Figure 2

Figure 2 shows a hierarchical data representation that includes multiple elements (structures 24a-j). Each element contains at least one "token"

Petition for Inter Partes Review of Patent No. 7,047,526

(abbreviated "T") and a corresponding "command key" (abbreviated "CK"). (*Id.* at 3:51-53.) Figure 2 also depicts a "command word translation table" (structure 20) that matches words within generic commands to tokens, which are shown as numerical values—for example, the generic-command word "watch" corresponds to the token "8." (*Id.* at 3:38-46 and Fig. 2.) These tokens are used to traverse the command parse tree and identify the element that's the "best match" for the received generic command. (*Id.* at 3:47-51.)

Tokens are matched with command-parse-tree elements based on the order of the words in the received generic command. (*Id.* at 4:3-7, 4:14-16.) Beginning at the highest level of the tree, the parser determines if the token corresponding to the first generic-command word is valid. (*Id.* at 4:3-13, Fig. 3.) If so, the parser traverses the next level of elements that depend from the already validated token to determine if the next token in sequence is valid at that level. (*Id.* at 4:13-16, 4:19-27.) Thus, for example, referring to Figure 2, if the first two words in the received generic command are "watch tcp," the parser determines if the token corresponding to "watch" (the number "8") is valid in element 24a, then determines if the token corresponding to "tcp" (the number "6") is valid in element 24b, which depends from token "8" in element 24a.

This process repeats until either no tokens remain, in which case the element corresponding to the final token is identified as the "best match," or the next token

Petition for Inter Partes Review of Patent No. 7,047,526

in the sequence is found to be invalid, in which case the element corresponding to the last valid token is identified as the "best match." (*Id.* at 4:27-36, 4:46-49.) The parser then uses the "command key" that corresponds to the "best match" element to identify a "prescribed command" to issue to a selected management program. (*Id.* at 3:51-54, 4:31-36, 4:49-51.) For example, referring to Figure 2, if the parser receives the generic command "watch tcp connections," it will identify element 24c as the "best match," and use the command key "3" to identify a "prescribed command" to issue to a selected management program.

B.   The state of the prior art

The idea of translating an input command from a high-level format to a tool-specific format was known for decades before the alleged invention of the '526 patent. (*See* Clark Decl. ¶¶ 20-27.) Since the 1970s, for example, UNIX "shell" programs accepted textual commands from a user and converted them into instructions for an operating system. (*See* Dennis M. Ritchie & Ken Thompson, The UNIX Time-Sharing System, 17 Communications of the ACM 365 (Jul. 1974) (Ex. 1012) at 7.) The UNIX shell and other programs converted user-input commands using a parser, which compared a statement to the rules of a language (known as its "grammar") and interpreted the statements. (*See* Brian W. Kernighan & Rob Pike, The UNIX Programming Environment (1984) (Ex. 1009) at 87.) Parsers are elemental to computer science and were well-known long before the

Petition for Inter Partes Review of Patent No. 7,047,526

'526 patent was applied for. (*See* Alfred V. Aho, et. al., Compilers: Principles,

Techniques, And Tools (1986) (Ex 1005) at 52-60. To cite just one example, in a

1974 paper, Ritchie and Thompson explained that the UNIX shell parses

commands and accompanying "arguments" (*i.e.*, parameters) that a user types into

the command line, noting that "[t]he Shell is a command line interpreter: it reads

lines typed by the user and interprets them as requests to execute other programs."

(Ex. 1012 at 7.)

The UNIX shell and related systems received commands input by a user

(including commands in a high-level command format), parsed the commands, and

issued appropriate procedure calls to external binary files. *Id.* Many other systems

did the same thing. As noted, for example, Digital Equipment Corporation's

"Digital Command Language," or "DCL," was used since the 1980s to translate

high-level commands entered by a user (*e.g.*, "PRINT/COPIES =

5   GROCERY.LIS," to print five copies of a grocery list file) into specialized

commands issued to external programs. (Ex. 1007 at 18-19, Section 3.3.1; *see also*

Clark Decl. at ¶ 25.)

The use of a "command parse tree" to parse sequences of tokens, as recited

in the'526 patent, was also obvious to persons of skill in the art by the time the

patent was applied for. Noam Chomsky's foundational work on formal languages

in the 1950's proposed a tree representation:

9

Petition for Inter Partes Review of Patent No. 7,047,526

> Every [finite-state, or regular] language can be represented in the form of a *tree*. At the root of the tree is a point from which all sentences start. Each word that can occur as the initial word in some sentence is represented by a branch leaving this initial point. At the end of the branch representing any particular word will be another set of branches representing each of the words that can follow the first. This process of arborization continues until every possible sentence is represented by some path through the tree; or, if the language contains an infinite set of sentences, the tree will continue indefinitely.

(Ex. 1003 at 4.) Skilled artisans therefore understood that any command language that has a finite number of possible commands can be represented as a tree.

Parsers of such languages process commands left-to-right, at each step consulting a data structure that represents the language's grammar, to see what command words are acceptable next choices. For finite command languages, these data structures are hierarchical representations of the language ("command parse trees"), as the '526 patent discloses. (Clark Decl. at ¶ 22-23.) As explained, for example, in U.S. Patent No. 6,134,709, applied for in June 1998, in its "Background of the Invention" section, "[t]he list of acceptable commands can be stored in a tree structure to reduce the space required to store all of the acceptable commands. The root nodes of the tree can contain acceptable first tokens in a command. . . . The tree can be searched to match the submitted command, token by token." (U.S. Patent No. 6,134,709 to Pratt, issued Oct. 17, 2000 (Ex. 1013) at 1:40-57; Clark Decl. at ¶ 23.)